# **Akenti** – A Distributed Access Control System

Srilekha S. Mudumbai, Mary R. Thompson, Gary Hoo
Abdeliah Essiari, Keith Jackson, William Johnston

http://www-itg.lbl.gov/Akenti
Imaging and Distributed Collaboration Group
Lawrence Berkeley National Laboratory, Berkeley, CA 94720

**Abstract:** DOE scientific resources – instruments, data, and collaborations – that are accessed via open networks require protection against unauthorized use. Akenti is designed to provide a flexible, easily managed mechanism, which strongly controls access to distributed resources, by widely distributed users.

## Introduction

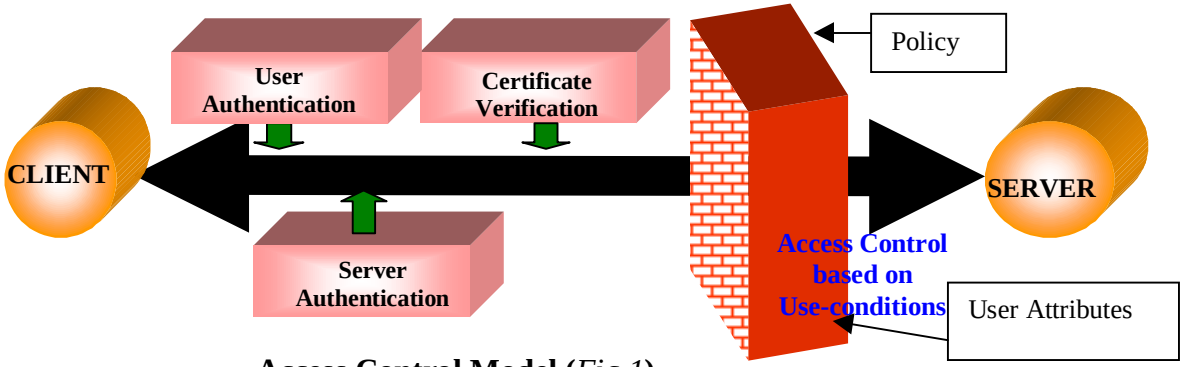
Akenti is an access control system designed to address the issues raised in allowing restricted access to distributed resources which are controlled by multiple stakeholders. The stakeholders are the people with authority to grant access to resources and may be both physically and organizationally remote from the resource. Akenti enables these stakeholders to remotely and securely create and distribute instructions authorizing access to their resources.

Access control is a means for enforcing an authorization policy. In a client-server architecture, the clients (on behalf of users) attempt to access resources that are controlled by servers. A priori authorization decisions govern which users may access which servers for what purposes and under what conditions. These decisions are reflected in an access control policy. Akenti makes access control decisions based on one set of digitally signed documents that represent the authorization instructions and another set that represent user attributes. Existing public-key infrastructure and secure message protocols provide confidentiality, message integrity, and user identity authentication, during and after the access decision process. (Fig. 1)

## Akenti Access Control Fundamentals

The resources that Akenti controls may be information, processing or communication capabilities, or physical systems such as scientific instruments. Access can be the ability to obtain information from the resource (as in "read" access), to modifying the resource (as in "write" access), or cause that resource to perform certain functions (as in changing instrument control set points). A network-based server acting as a proxy for the resource typically provides remote access to a resource. A user gains access to the resource via a client program. The client, therefore, has to participate in a series of authentication and verification steps before gaining access to the resource as follows:

- In an initial two-way process, the client authenticates the server, after which it accepts information from that server. Following this, the server authenticates the client by checking the user's identity credentials.
- There is set of use-conditions imposed on the resource by the stakeholders in the form of digitally signed documents ("certificates"). If the user satisfies the use-conditions by possessing attributes that match the use-conditions then the client is permitted specified actions (i.e. read, modify etc.) on the resource.




**Access Control Model (*Fig 1*)**

Authorization is the granting of rights, by the owner or controller of a resource, for others to access that resource. Authorization is enforced at the two levels itemized above. The initial identity authorization that is required to gain access to the server is accomplished by checking the user's identity. This identity is represented by a Public Key Infrastructure (PKI) X.509 certificate. If the identity certificate was signed by a Certificate Authority trusted by the server, the initial authorization succeeds.

The second level of authorization verifies that the user satisfies the use-conditions that are imposed by the stakeholders on the resource. The stakeholders are responsible for establishing policy at the resource level. They do this through a combination of authority files and use-condition certificates. The user satisfies the use conditions by presenting the appropriate attribute certificates.

## Components of the Model

***Identity (X.509) certificates***

X.509 Certificate Authorities (CAs) are used to issue and digitally sign the identity certificates for the user. (Fig. 2) These certificates associate an entity's name with a public-key, and bind them together through the digital signature of the CA. They are stored in Lightweight Directory Access Protocol (LDAP) servers, from which the policy engine uses them to verify issuers and user's identities. A subject's identity is verified if its X.509 certificate is signed by one of the trusted Certificate Authorities listed in the authority files. The user manages his certificates with standard programs that implement PKI, such as the Netscape browser.

***Use-condition certificates***

These are signed documents, remotely created and stored by resource stakeholders that specify the conditions for access to the resource. They include

- Combinations of required attributes and values
- Name of the resource
- Permitted actions
- Identities to be trusted to issue related attribute certificates
- Certificate Authorities (CA) to be trusted to verify the user and issuer identities
- Signature of the use-condition issuer

***Attribute certificates***

These are signed documents, remotely created and stored that certify that a user possesses a specific attribute, (for example, membership in a named group, completion of a certain training course, or membership in an organization). They include

- Attribute name

**Certificate Content:**

```
Certificate:
    Data:
        Version: v3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer: CN=IDCG-CA, OU=ICSD, O=Lawrence Berkeley National Laboratory, C=US
        Validity:
            Not Before: Fri Aug 29 11:25:05 1997
            Not  After: Sat Feb 20 10:25:05 1999
        Subject: CN=Srilekha Mudumbai, OU=ICSD, O=Lawrence Berkeley National Laboratory,
        Subject Public Key Info:
            Algorithm: PKCS #1 RSA Encryption
            Public Key:
                Modulus:
                    00:9c:23:9e:55:bf:50:9c:99:76:7d:02:fe:77:3e:b9:ec:90:
                    90:8c:a5:e9:0f:99:1a:db:1f:c7:db:9d:c2:36:02:c4:c9:bc:
                    14:94:9e:06:a0:a5:ba:26:0f:3d:2e:b3:d5:b7:a7:cc:19:02:
                    d4:1c:d5:09:c4:67:f0:f2:e0:bd:9b
                Public Exponent: 65537 (0x10001)
        Extensions:
            Identifier: Certificate Type
                Critical: no
                Certified Usage:
                    SSL Client
            Identifier: Authority Key Identifier
                Critical: no
                Key Identifier:
                    09:07:1d:ab:52:ef:c1:5a:6b:33:b9:0b:94:f2:e5:ed:f9:96:
                    e0:fb
    Signature:
        Algorithm: PKCS #1 MD5 With RSA Encryption
        Signature:
            91:52:28:a1:48:97:5c:e8:51:3c:c1:83:4c:7d:b4:bb:09:cd:ae:ec:82:
            e8:71:2d:2a:f8:73:a8:55:fd:f1:50:94:ee:37:1b:5c:10:da:47:23:be:
            0e:44:81:c0:3e:1d:65:9d:2a:1e:6f:05:16:d6:00:46:27:78:57:d6:58:
            9e:7f:5d:b1:c1:4e:12:1b:39:2a:53:2a:94:a6:2b:1b:a6:e5:ed:a6:e3:
            4a:9c:de:11:15:f6:c5:20:9c:d7:bc:ae:77:8c:12:bc:c0:4b:38:58:06:
            11:a0:01:c2:70:8b:b6:75:4d:0d:15:48:ab:c8:b6:4b:da:7b:3b:91:c3:
            06:ba
```

**X509 Certificate (*Fig 2*)**

- Name space of the attribute
- Value
- Auxiliary information (e.g. time interval)
- Subject (User) and its trusted CA
- Issuer and its trusted CA
- Signature of the attribute certificate issuer

***Authority file***
This is information stored on the resource server associated with each resource. It consists of relatively static information that specifies who can provide access information (i.e. defines the stakeholders), where to look for access-control information, and what CAs to trust.

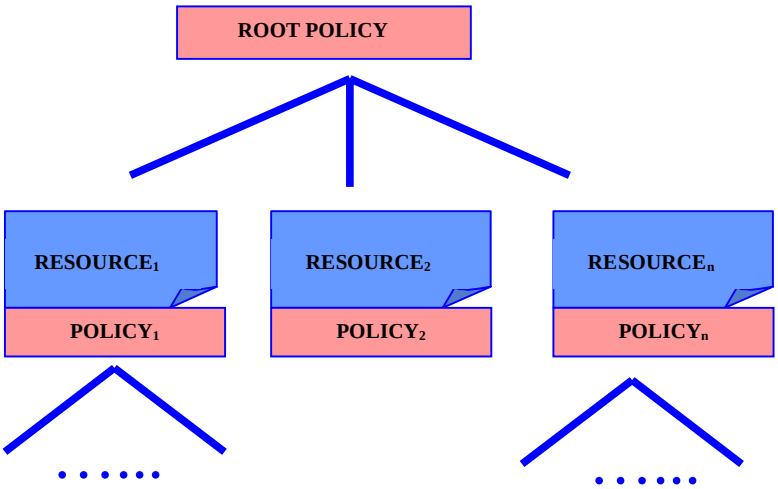
The use condition certificates specify the attributes and values that a user must satisfy. If corresponding attribute certificates can be found for the user, then the use-conditions are satisfied, and the user's client is allowed access. If not, access is denied.

- The extent to which policies may be stated in terms of general rules

A typical policy model is hierarchical (Fig. 3). Akenti's current model allows both individual-based policies (stakeholders) and an overall policy. For example, a resource administrator would be responsible for defining a root policy for the system. The root policy establishes trusted CA's and any global access restrictions. Policies for individual resources or groups of resources are expressed in terms of pointers to use-conditions that are designed by the stakeholders.

## Implementation Status

***Web Server***
Currently, Akenti is used by an Apache Web Server that implements a hierarchical policy model (Fig. 3) and uses SSL (Secure Socket Layer) in order to provide confidentiality and integrity of communication. The "Akenti policy engine" interfaces to the Apache server to provide Akenti's access control. This server is being used to provide access control of experimental results that are generated by members of the DOE 2000 Diesel Combustion Collaboratory [2].




**A Hierarchical Model (*Fig 3*)**

## Access Control Policy Model

Various policy models must be supported to accommodate different environments. Policy models differ in terms of

- The level at which the authorization decision is made (i.e. is the policy set only at the top level or is delegation of authority allowed, where decisions are made by stakeholders for specific resources.)
- The ways in which users and/or resources are grouped together for purposes of common handling

***CORBA ORB***
Akenti access control has been added to a CORBA ORB (Orbix including the SSL communication protocol). The DOE 2000 Materials MicroCharacterization Collaboratory

[3] is planning to use this ORB to protect a microscope control server. An ORB defines a callout function that is automatically called whenever a request is initiated. This callout hook is used to call the Akenti policy engine. The result of this call either permits the requested access or denies it, depending on whether or not the user satisfies the use conditions.

***Use condition and Attribute certificate creation***

Use-condition and attribute certificate can be created using a graphical interface, implemented using Java JDK1.1, Java Workshop and JDBC.

## Future Work

Future work will include deploying a mobile agent system with Akenti access control; integrating Akenti access control with Globus resource management system in a secure compute server and a network bandwidth broker; and integrating Akenti with Sandia Lab's PRE/CORBA remote server invocation system.

## More Information

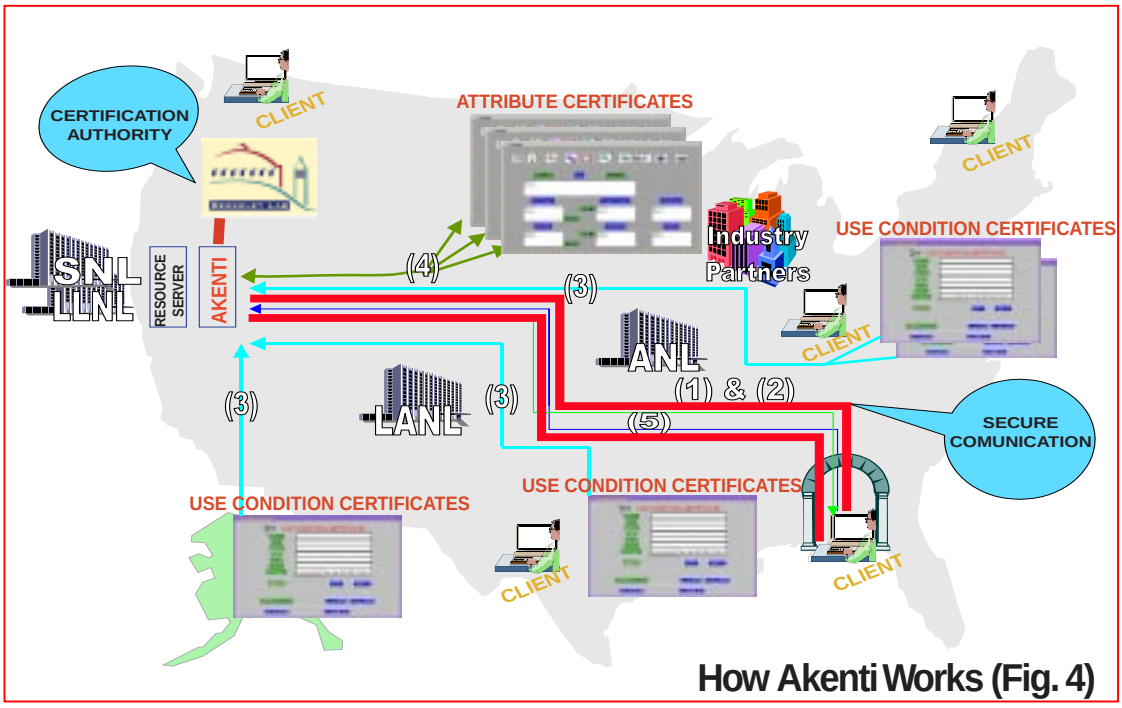
[1] A detailed view of the project can be obtained from the following web site
http://www-itg.lbl.gov/Akenti/

[2] An early deployment of Akenti is being used in support of the DOE 2000 DCC
http://www-collab.ca.sandia.gov/

[3] DOE 2000 Materials MicroCharacterization Collaboratory
http://tmp.amc.anl.gov/MMC/

## Acknowledgement

The work described above is supported by the U. S. Dept. of Energy, Energy Research Division, Mathematical, Information, and Computational Sciences office under contract DE-AC03-76SF00098 with the University of California.




***Steps in the Access Control Process***
The access control process involves five steps. (See Fig. 4): (1) The client (user) authenticates the server; (2) the server authenticates the client by reference to a trusted CA; (3) the policy engine gathers use-condition certificates from the stakeholders' servers; (4) the attributes required by the use-conditions are verified by obtaining attribute certificates from trusted directories, and; (5) if the user satisfies all the requirements, a secure connection is established between the client and the server. Otherwise the policy engine denies access to the server's resource.